

Introduction to Computers and Programming

Prof. I. K. Lundqvist
kristina@mit.edu , 33-318

Readings: Ada/Mindstorms 2.0 User's Guide and Reference Manual
http://www.usafa.af.mil/dfcs/Ada_Mindstorms_manual.htm

Lecture 2
Sept 15 2004

Mud from yesterday

- How does Ada interface other languages?
 - Annex B ALRM
- What makes Ada more robust than e.g. C++ or Java
 - Ada enforces good programming practices
 - Exception handling
 - Type safety...
 - Good engineering is the key

Today

- Lego Mindstorms Logistics
- RCX 2.0
- Ada/Mindstorms Basics
- Programming Examples

3

PRS

Have you used Lego Mindstorms before?

1. *No* previous experience
2. I have *some* experience using Lego Mindstorms
3. I feel *confident* in building Lego Mindstorms robots and/or programming them

4

Lego Mindstorms Logistics

- Teams of 3 per 1 Lego kit
 - Collaborate on the pset questions that involve Lego Mindstorms
- Instructions on taking ownership of the kits will be distributed via email
- Handle the kits and parts responsibly
 - Kits will be used in upcoming courses (16.00, 16.35 and 16.unified)

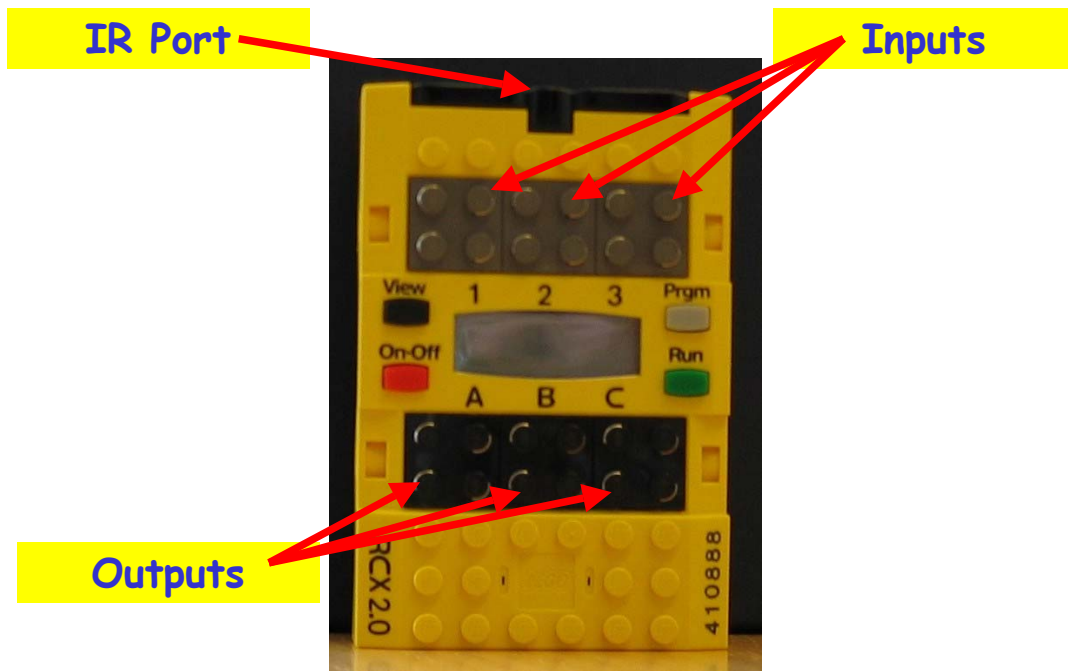
5

The Lego Mindstorms Robotics Invention System 2.0

- RCX™ Microcomputer
- CD-ROM Software
- USB Infrared Transmitter
- 718 pieces, including:
 - 2 Motors
 - 2 Touch Sensors
 - 1 Light Sensor
 - (+1 Rotation Sensor)



RCX 2.0

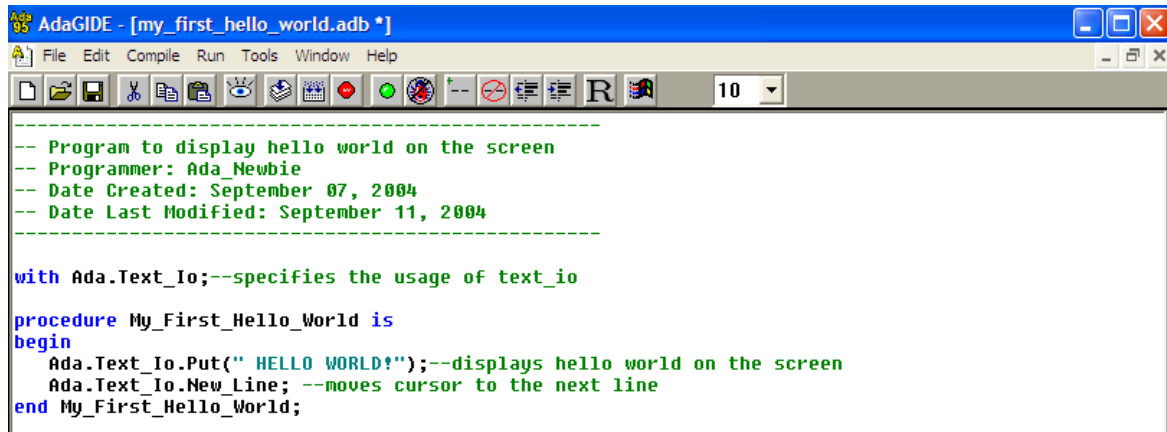


7



8

Ada – Hello World



The screenshot shows the AdaGIDE IDE window titled "AdaGIDE - [my_first_hello_world.adb *]". The menu bar includes File, Edit, Compile, Run, Tools, Window, and Help. The toolbar contains icons for file operations, editing, and execution. The code editor displays the following Ada code:

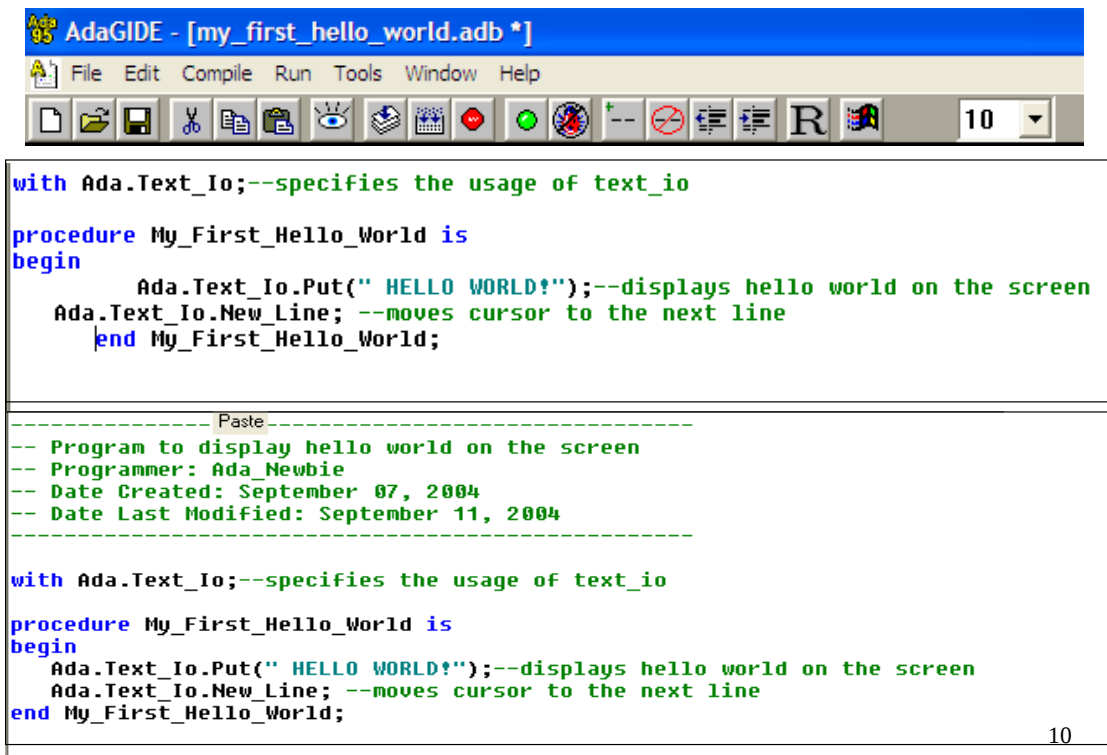
```
-- Program to display hello world on the screen
-- Programmer: Ada_Newbie
-- Date Created: September 07, 2004
-- Date Last Modified: September 11, 2004

with Ada.Text_IO;--specifies the usage of text_io

procedure My_First_Hello_World is
begin
  Ada.Text_IO.Put(" HELLO WORLD!");--displays hello world on the screen
  Ada.Text_IO.New_Line; --moves cursor to the next line
end My_First_Hello_World;
```

9

Ada – Reformatting



The screenshot shows the AdaGIDE IDE window titled "AdaGIDE - [my_first_hello_world.adb *]". The menu bar includes File, Edit, Compile, Run, Tools, Window, and Help. The toolbar contains icons for file operations, editing, and execution. The code editor displays the following Ada code, which has been reformatted with consistent indentation:

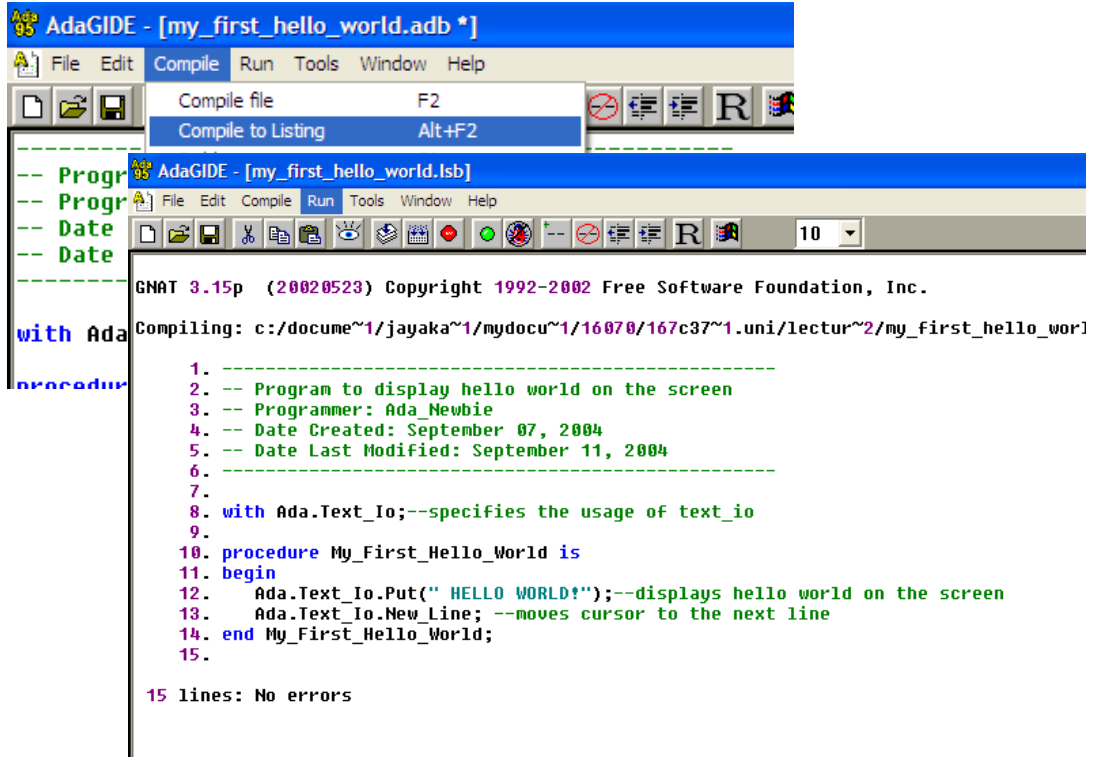
```
-- Program to display hello world on the screen
-- Programmer: Ada_Newbie
-- Date Created: September 07, 2004
-- Date Last Modified: September 11, 2004

with Ada.Text_IO;--specifies the usage of text_io

procedure My_First_Hello_World is
begin
  Ada.Text_IO.Put(" HELLO WORLD!");--displays hello world on the screen
  Ada.Text_IO.New_Line; --moves cursor to the next line
end My_First_Hello_World;
```

10

Ada – Compile to Listing



```
AdaGIDE - [my_first_hello_world.adb *]
File Edit Compile Run Tools Window Help
Compile file F2
Compile to Listing Alt+F2

AdaGIDE - [my_first_hello_world.lsb]
File Edit Compile Run Tools Window Help
10

GNAT 3.15p (20020523) Copyright 1992-2002 Free Software Foundation, Inc.
Compiling: c:/docume~1/jayaka~1/mydocu~1/16070/167c37~1.uni/lectur~2/my_first_hello_wor]

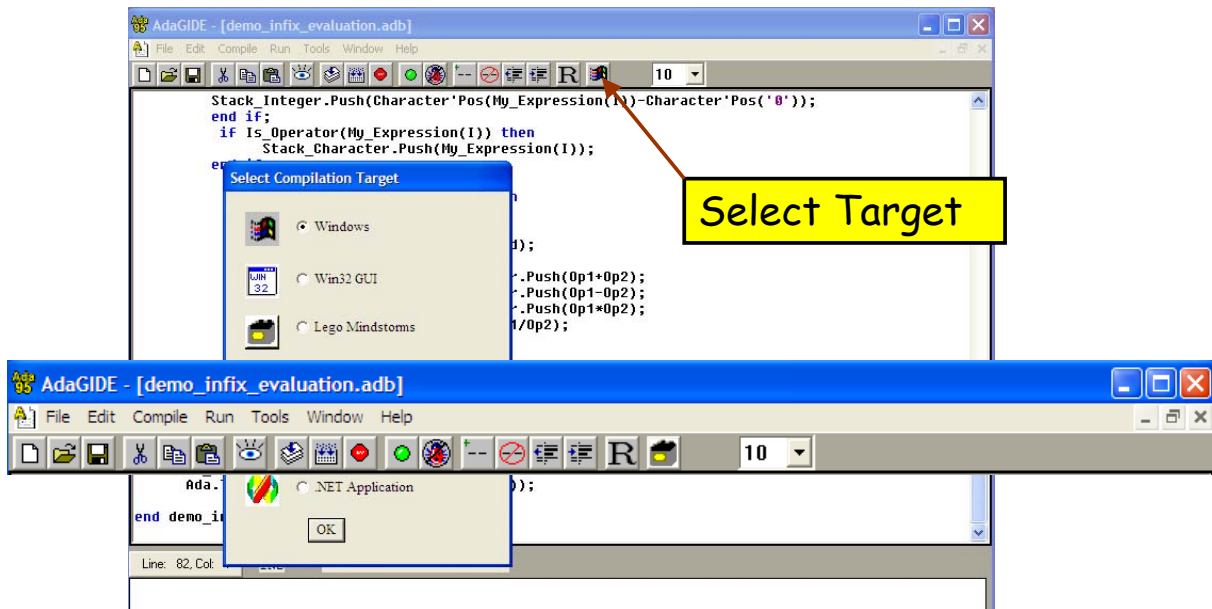
1. -----
2. -- Program to display hello world on the screen
3. -- Programmer: Ada Newbie
4. -- Date Created: September 07, 2004
5. -- Date Last Modified: September 11, 2004
6. -----
7.
8. with Ada.Text_IO;--specifies the usage of text_io
9.
10. procedure My_First_Hello_World is
11. begin
12.   Ada.Text_IO.Put(" HELLO WORLD!");--displays hello world on the screen
13.   Ada.Text_IO.New_Line; --moves cursor to the next line
14. end My_First_Hello_World;
15.

15 lines: No errors
```

Ada/Mindstorms 2.0

- Ada -> NQC -> Lego Bytecode
 - Supports a small subset of Ada95
- Lego package
 - Subprograms
 - Types

Ada Mindstorms

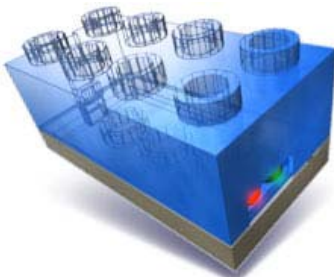


http://www.usafa.af.mil/dfcs/Ada_Mindstorms_manual.htm

13

Sensors

- Touch
- Rotation
- Light



14

Sensor Ports

```
type Sensor_Port is (Input_1, Input_2, Input_3);
```

- Type
- Mode

```
type Sensor_Type is  
  (Type_Touch,      -- a touch sensor  
   Type_Temperature, -- a temperature sensor  
   Type_Light,       -- a light sensor  
   Type_Rotation);  -- a rotation sensor
```

```
type Sensor_Mode is (  
  Mode_Raw,      -- number from 0 to 1023  
  Mode_Bool,     -- 0 or 1  
  Mode_Edge,     -- counts number of Boolean transitions  
  Mode_Pulse,    -- counts number of Boolean periods  
  Mode_Percent,  -- number from 1 to 100  
  Mode_Celsius,  -- degrees C  
  Mode_Fahrenheit, -- degrees F  
  Mode_Rotation); -- counts rotations
```

Creating Sensor Ports

- `Set_Sensor_Type(Sensor => Sensor_1,
 Mode => Type_Touch);`
 - `Set_Sensor_Mode(Sensor => Sensor_1,
 Mode => Mode_Edge);`
- Or**
- `Config_Sensor(Sensor => Sensor_1,
 Configuration => Config_Edge);`

Output Ports

```
type Output_Port is (Output_A, Output_B, Output_C);
```

- Mode
- Direction
- Power Level

```
type Output_Mode_Type is (Output_Mode_On, Output_Mode_Off);
```

```
type Output_Direction is (Output_Direction_Forward,  
    Output_Direction_Reverse, Output_Direction_Toggle);
```

```
type Power_Type is range 0..7;  
Power_Low      : constant Power_Type := 1;  
Power_Half     : constant Power_Type := 4;  
Power_High     : constant Power_Type := 7;
```

17

Configuring Output Ports

- Set_Output_Mode(Output => Output_1,
 Mode => Output_Mode_On);

or

- Output_On(Output => Output_1);

18

Example

- What is the rover doing?

19

Example 1/6

```
with Lego;
```

```
use Lego;
```

```
procedure Demo_Basic_Rover is
```

```
  Left  : constant Integer := 0;
```

```
  Right : constant Integer := 1;
```

```
  Drive : constant Output_Port := Output_A;
```

```
  Steer : constant Output_Port := Output_B;
```

```
  Straight : constant Sensor_Port := Sensor_1;
```

```
  Bumper   : constant Sensor_Port := Sensor_2;
```

20

Example 2/6

```
procedure Initialize_Rover is  
  begin  
    Config_Sensor( Sensor => Straight,  
                  Config => Config_Touch);  
    Config_Sensor( Sensor => Bumper,  
                  Config => Config_Touch);  
    Output_Power( Output => Drive,  
                  Power  => Power_High);  
    Output_Power( Output => Steer,  
                  Power  => Power_Low);  
    Steer_Center;  
  end Initialize_Rover;
```

21

Example 3/6

```
procedure Steer_Left is  
  begin  
    Output_On_Forward(Output => Steer);  
    Wait(Hundredths_Of_A_Second => 25);  
    Output_Off(Output => Steer);  
  end Steer_Left;  
  
procedure Steer_Center is  
  begin  
    Set_Output_Direction(Output => Steer,  
                          Direction => Output_Direction_Toggle);  
    Output_On(Output => Steer);  
    while (Get_Sensor_Value(Sensor=>Straight) =0) loop  
      Wait(Hundredths_Of_A_Second => 1);  
    end loop;  
    Output_Off(Output => Steer);  
  end Steer_Center;
```

22

Example 4/6

```
procedure Turn ( Direction : in      Integer ) is  
begin  
    if Direction = Left then  
        Steer_Center;  
        Steer_Left;  
    else  
        Steer_Center;  
        Steer_Right;  
    end if;  
end Turn;
```

23

Example 5/6

```
procedure Go_Forward is  
begin  
    Output_On(Output => Drive);  
end Go_Forward;  
  
procedure Forward (Tenths_Of_A_Second : in Integer ) is  
begin  
    Output_On_Forward(Output => Drive);  
    Wait(Hundredths_Of_A_Second =>  
        Tenths_Of_A_Second * 10);  
    Output_Off(Output => Drive);  
end Forward;
```

24

Example 6/6

begin

Initialize_Rover;

Forward(20);

Turn(Left);

Forward(5);

Steer_Center;

Back(20);

Turn(Right);

Back(5);

Steer_Center;

Forward(20);

end Demo_Basic_Rover;